

July-August 1989 Edition

RE RUN

RUN Programs on Disk

For the C-64 and C-128



Plus: Extra Bonus Programs!

Introduction

July–August '89 ReRUN

WRITING THIS COLUMN is the most exciting part of compiling the ReRUN booklet and disk every other month. It gives me an opportunity to pass along my personal opinion and insight on the programs found on each disk.

Some of you may have heard that I was recently appointed technical editor for *AmigaWorld*. Don't worry. While I'm excited to be a member of the *AmigaWorld* staff, I'm equally dedicated to the continued support of ReRUN customers as the ReRUN technical editor. Rest assured that not a day goes by when I don't work (and occasionally play) on the C-64 and C-128. Furthermore, I remain available at ReRUN by both letter and toll-free telephone number to answer your questions, fix, replace or check your ReRUN disks and otherwise provide technical assistance.

Like most ReRUN users, I fully appreciate software written for Commodore 8-bit computers that takes maximum advantage of the capabilities of those machines. That type of software is exemplified in this issue's first program, RUN Shell. Written by John Ryan, *RUN's* contributing editor, RUN Shell is a pure C-64 and C-128 machine language disk utility that was originally conceived as an alternative to the 1571 DOS Shell program that Commodore bundles with both the C-128D and 1571 disk drive. One nice feature John managed to build into it makes it possible for the program to determine whether it's running on a C-64 or a 128 (in 40- or 80-Column mode) and configure itself accordingly.

RUN Shell also provides an option for formatting your 1764, 1700 and 1750 RAM expanders for use as RAM disks for file and disk transfers. Complete 1581 support is yet another nicety offered by RUN Shell. I encourage everyone who has a 1581 connected to his system to keep a copy of RUN Shell on a 1581 disk for rapid file-copying between the 1581 and your 1541 or 1571 disk drives. I'm sure you'll find yourself relying on RUN Shell as one of your most frequently used disk utilities.

A–Z Scramble is the next program on this disk. For the C-64, it allows up to five players to alphabetize from one to five grids of

random letters. It's a mentally stimulating game for computerists of all ages.

If you're a C-128 user who finds A-Z Scramble a little too easy, try Cubix I. Written by Leonard Morris, this is a three-dimensional game in which you must properly orient and position six sections to create a large cube on the screen. The graphics and techniques used are first-rate; it's certain to keep C-128 computerists busy for hours.

A Better Mousetrap, for both the C-64 and the C-128, prints the X (horizontal) and Y (vertical) coordinates of an invisible screen pointer as you move the 1351 mouse. The significance of this technique is obvious to programmers seeking methods to develop 64 and 128 programs that utilize the 1351 mouse.

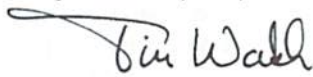
Bob Kodadek, author of *RUN's* Checksum and many other fine programs, wrote Label Base, the next program on this disk. Like *RUN Shell*, Label Base automatically determines whether it is operating on a C-64 or C-128, and in 40- or 80-Column mode, and adjusts itself accordingly. As its name implies, Label Base is a mini-database for creating and printing lists of names, addresses and telephone numbers. A professional menu design, with options for editing and alphabetizing entries and for saving and loading files, makes Label Base a pleasant, easy-to-maintain program.

Nothing makes a computer program more intriguing than screen animation. Memory in Motion, a C-64 program, brings animation to the board tiles found in the traditional game of Concentration. Along the same lines of unique entertainment is Thoroughbird Racing, an exciting C-128 ostrich racetrack simulator. Written by the recently retired physician, Hugh McMenamin, Thoroughbird Racing lets you and your friends place bets on the big birds at a pari-mutuel window before each race, and a conscientious woman behind the counter keeps track of each player's winnings and losses.

Fast-Food Frenzy, a C-64 squirt-'em-up, lets you aim, squeeze and fire a stream of ketchup at incoming bags of fries and burgers. I can't recommend this program to anyone suffering on a low-calorie diet, because it never fails to cause hunger pangs whenever I play it.

Two bonus programs round out this edition of ReRUN. The first, Storybook 64, is a building block-type program that lets intermediate level Basic programmers create a computerized storybook, complete with graphic screens. Budget Plus, another Leonard Morris original, gives C-128 users a new way to do their budgets. Give it a try. I think you'll find that it not only works well, but that it's packed with more bells and whistles than you can imagine.

That's it for now. As you've come to expect, this edition of ReRUN is one of the best collections of useful and entertaining C-64/C-128 software you'll find anywhere—and I intend to keep it that way for you.



Jim Walsh
Technical Editor
ReRUN

Directory

PAGE	DOCUMENTATION	DISK FILENAME	FILE TYPE
1	RUN SHELL	*MENU 128 _____	BASIC
		MENU 64 _____	BASIC
		RUN SHELL.BOOT _____	BASIC
		RUNSHELL.HEX _____	BASIC
		RUN SHELL _____	ML
7	A-Z SCRAMBLE	A-Z SCRAMBLE _____	BASIC
8	CUBIX 1	*CUBIX 1 _____	BASIC
		*ML-CUBIX 1.HEX _____	BASIC
		*ML-CUBIX 1 _____	ML
9	A BETTER MOUSETRAP	MOUSETRAP 64 _____	BASIC
		*MOUSETRAP 128 _____	BASIC
13	LABEL BASE	LABEL BASE _____	BASIC
17	MEMORY IN MOTION	MEMORY MOTION _____	BASIC
17	THOROUGH BIRD RACING	*THOROUGH BIRD _____	BASIC
		*THOROBIRD.SPRITE _____	BASIC
		*SPRITES _____	ML
18	FAST-FOOD FRENZY	KETCHUP ATTACK _____	BASIC
		KETCHUP.HEX _____	BASIC
		KETCHUP _____	ML
19	£STORYBOOK 64	STORYBOOK 64 _____	BASIC
		DEMO PART 1 _____	BASIC
		DEMO PART 2 _____	BASIC
		BOOK COVER MPIC _____	ML
		FRONT PIC PIC _____	ML
		PLAUSERS _____	ML
		FDJAGUAR _____	ML
20	£BUDGET PLUS	*BUDGET PLUS 128 _____	BASIC

* — C-128 mode only

£ — Bonus program

Before you run a program, carefully read the documentation that pertains to it.

How To Load

LOADING FROM MENU

To get started, C-64 users should type `LOAD "MENU 64",8` and press the return key. When you get the Ready prompt, the menu is loaded and you should type `RUN` to see a list of the programs on your disk. C-128 users need only press the shift and run-stop keys. When all the programs are displayed on the screen, you can run the one you select by pressing a single key.

LOADING FROM KEYBOARD

If you do not wish to use the menu program, follow these instructions.

C-64: To load a C-64 program written in Basic, type: `LOAD "DISK FILENAME",8` and then press the return key. The drive will whirl while the screen prints `LOADING` and then `READY`, with a flashing cursor beneath. Type `RUN` and press the return key. The program will then start running. To load a C-64 program written in machine language (ML), type: `LOAD "DISK FILENAME",8,1`

C-128: All C-64 programs can be run on the C-128 as long as your computer is in C-64 mode. All C-128 programs are clearly labeled on the directory page. Your C-128 *must* be in C-128 mode to run these programs. To load a C-128 mode program, press the F2 key, type the disk filename and then press the return key. When the program has loaded, type `RUN`.

MAKING COPIES OF ReRUN FILES

Many programs on your ReRUN disk have routines that require a separate disk onto which the program writes or saves subfiles. To use these programs, you must first make a copy of the original program onto another disk that has enough free space on it to hold these newly written subfiles.

It's simple to make a copy of a Basic program. Just load it into your computer as outlined above, and then save the program back onto a separate disk that has plenty of free space for extra files.

Copying an ML program is not so simple. You cannot simply load and save an ML program; you'll need to use a disk-backup utility program, such as the one on your Commodore Test Demo disk.

RUN it right: C-64; C-128 (in both 40- and 80-Column modes);
1541, 1571 or 1581 disk drive

RUN Shell

By John Ryan

WITH THE INTRODUCTION of the 1581 disk drive and RAM expansion modules, I've developed a DOS utility compatible with both the C-64 and C-128 and all three disk drives. Called RUN Shell, it's the newest and most powerful DOS shell to date.

RUN Shell is a file-maintenance and disk-archiving utility that runs in both 64 and 128 modes. It also supports both 40- and 80-column output; Burst mode; all three Commodore RAM expansion modules, to a limited degree; and partitioning and subdirectory access on the 1581 drive.

GETTING STARTED

Activate RUN Shell by using either Menu 64 or Menu 128. If you intend to employ a RAM expansion module, be sure it's installed before power-up.

The program can be loaded in either 64 or 128 mode, but don't try to load it in one mode and switch to another, because it configures itself to fit the particular machine. You may, however, toggle between 80- and 40-Column modes.

Although it's a stand-alone program, RUN Shell will operate with many machine language programs that don't occupy memory from \$1300 to \$5000. However, due to the way it configures itself under the Basic of both machines, it won't operate with many Basic programs. This limitation affects users of RAM expansion modules, but more on that later.

Almost all of the options available in RUN Shell bring to the shell window a directory from which you can select files to work on. To choose a file for some operation, use the cursor keys to move the directory arrow to your selection. Then highlight the filename by pressing the space bar. If you change your mind, just deselect the file with the F5 key.

When you're finished highlighting all the files you want for any

particular operation, you activate the operation by pressing the F7 key. In many cases (but not all), a Y/N prompt will ask you to verify your selection before pressing F7.

You can usually abort to the main menu by pressing the stop key. If not, there'll *always* be a Y/N prompt later to abort the operation.

There will be instances where RUN Shell expects a keypress from you to continue an operation (when swapping disks during a file copy, for example). The rule here is to press the space bar to continue or the stop key to abort to the main menu.

To exit RUN Shell, press the F1 key, and press it again to re-enter. C-64 owners must reinitialize the program after using the run-stop/restore combination by entering SYS 4864.

THE OPTIONS

Disk Setup—You can configure RUN Shell to operate under two different logical drives (logical meaning that device numbers and drives are handled internally with the shell program). As you'll see, these logical drives—drive A and drive B—can be set up for any combination of disk or RAM drives, as well as device numbers 8–11. To keep things straight, I'd suggest that you set the lower device number to drive A and the higher device number (including RAM drives) to drive B.

You can also configure RUN Shell for single-drive operation by setting both the A and B drives to the same device number.

Rename a File—With the Rename utility, you can rename all the files in a directory. The directory is displayed in the shell window, and you select and highlight the filename you want to change by using the cursor keys and space bar as described above. When the highlight bar appears, enter a new filename that's no longer than 16 characters; then press return and continue the operation. When you're ready to activate all the new names for the directory, press the F7 key.

Scratch a File—This option operates much like Rename. Use the cursor keys and space bar to select and highlight filenames you wish to scratch. When all selections are made, press F7. The F5 key deselects a file if you change your mind after highlighting, and the stop key aborts to the verification prompt. Neither the Rename nor Scratch option works with RAM drives.

Format a Disk—Formatting is one of the most frequently used options in RUN Shell. First, the Format routine attempts to read a disk header from the selected drive. If an error occurs, it assumes

the disk hasn't been formatted before and prompts for a header name and two-letter ID. If the disk has been formatted before, the header name appears in the input window. Then you can reformat with the same header name by pressing the return key, or you can type in a new name, with or without a two-letter ID.

If the drive selected is a RAM device, you needn't enter a header name, but you do have to specify the type of device at the prompt. *All* modules must be formatted before being used as a logical drive or you'll get a RAM Disk Full error message.

Clean Up a Disk—This is a collect operation, and no verification is involved, so be sure the disk is in the proper drive.

Copy File—Using this option, you can copy files (except relative files) between any two disk drives other than a RAM expander (see RAM, below). Just highlight the files you want to copy and then press F7.

Commodore 64 users are limited to copying files of 90 disk blocks or less, while C-128 users can go up to 180 blocks. Use the Copy Disk routines to copy disks with larger file sizes.

To copy files from a 1581 subdirectory, you must first open the subdirectory with the Directory routine. With a single 1581 drive, you can copy only 90/180 disk blocks at a time, because RUN Shell won't detect previously opened directories after each disk swap. (RAM expansion users have a way around this limitation.)

Copy Disk—The Copy Disk option completely reproduces a disk. It works best with two drives, although it can be used with one. If you have a 1581, keep in mind that trying to single-copy 800K of data with a C-64 takes about 26 disk swaps (16 for the C-128), so use the Copy File routine when appropriate. It's very important that 1541/1571 users format the target disk either before entering the Copy Disk routine or while in the routine to ensure faithful data reproduction; 1581 users must format their disks beforehand.

To contrast C-64 and C-128 copying times, the C-64 option single-copies a 1581 disk in about 50 minutes and a 35-track disk in about 25 minutes. The C-128, with its Burst protocol, single-copies the same disks in about 12 minutes and a little over seven minutes, respectively. These times are slightly lower in dual-drive operation.

You can't copy from a 1581 drive to a 1541/71, because of the different formats involved.

Create Dir/RAM—These two options are for 1581 and RAM expansion module owners only. With Create Dir, you can create partitions and subdirectories on the 1581 disk drive more easily

than with the utility supplied on the demo disk. Just select the starting track of your partition with the + or - key (pressing the space bar to lock in the selection); then use + or - to scroll through the size of directory desired, and RUN Shell will automatically make the calculations for creating the partition. Press the space bar again to lock in the block size, and then answer the prompts as they appear.

To create a subdirectory from the partition, answer the formatting prompts as they come on-screen. You don't need to format a subdirectory within this option. You can format the partition by opening it within the Directory option and then using the Format option. Formatted partitions will always be 40 blocks less than your selected number in order to accommodate the partition directory.

Use this option only on new disks, as the partition routine overwrites any data already in the selected tracks of the disk.

You can create nested directories (directories within directories) as long as each encompasses at least three tracks and doesn't fall on the directory track (the first track of the directory) of the previous partition. Consider the following mock directory:

Main subdirectory #1: tracks 1-20

Subdirectory #2: tracks 2-5 (within main)

Subdirectory #3: tracks 6-18 (within main)

Notice that all the directories are at least three tracks long (minus nested subdirectories), and none starts on the first track of the other.

Once a directory is in place, you must access the "root" directory by pressing the stop key (see Directory, below). Then you must reopen the directory in which you want to nest the new subdirectory before you access the Create Partition option. I'd advise you to play around with this option before attempting any serious partitioning, and to refer to the 1581 user's guide for more information on partitions and subdirectories.

RAM—This option for owners of RAM expansion modules is a supplement to Commodore's RAMDOS utility, not a replacement. It includes three suboptions and requires that your RAM expander must be previously formatted.

The System to Expansion suboption lets you load disk files into the RAM drive. When RUN Shell asks you which logical drive to use, always select the logical RAM drive as the target (to load to) and a disk drive as the source (to load from). Then highlight the desired files and press F7 to initiate the load. *Don't* attempt to load files containing more than 90 disk blocks on the C-64 or 180 blocks on the C-128, or you might corrupt the computer's memory. You'll

notice that the number of blocks the drive has free corresponds to the bytes available in each RAM expansion module (less the overhead directory space that's maintained in each module).

The second suboption loads files from the RAM drive into system memory. Here again, select whichever logical drive is assigned to the RAM drive. (This may sound confusing, but you'll get the hang of it.) With this suboption, you never load from a disk drive.

Plan on this operation handling only sequential files and machine language programs. Since Basic is reconfigured to accommodate RUN Shell, many Basic programs won't work properly with it.

With this option, you don't have to load a word processor to examine a sequential file. Just load the file into the RAM drive and download it into system memory; then, using the cursor keys, scroll through it in the directory window. If you have a machine language file that doesn't occupy the same memory area as RUN Shell (\$1300-\$5000), you can download it, then exit the Shell and SYS to the program.

Machine language programmers might like to know that the Power Assembler can function when RUN Shell is active, since Basic memory is moved up to location \$5000. This may be true for other assembler systems, as well.

The third RAM suboption lets you "dump" all RAM directory files to disk and provides a handy way of copying files from one subdirectory to another. Just open a subdirectory on your 1581 drive and select Disk to Expansion to load the files into memory. Then close the subdirectory (see Directory, below), open the target subdirectory and select Expansion to Disk.

Select the logical RAM drive as the source and the logical disk drive as the target for this suboption. There's no need to select files here, since the entire RAM directory is dumped to disk.

This suboption will appeal to those who want to make multiple copies of disk files without a multitude of disk swaps, and it's handy for cataloging disks. Normally, grouping similar programs together on a single disk involves dozens of disk swaps and a big headache. With this RUN Shell feature, you can selectively load files from several dozen disks into the RAM drive, then dump them all to a single disk.

Disk Log—The Disk Log option reports the starting and ending addresses of a selected file, as well as the total number of bytes it contains. It handles only one filename at a time, and, with sequential files, only the total number of bytes is significant, since the starting

and ending addresses reported are those used by RUN Shell.

Directory—This option displays a disk directory in the directory window, and, if the device is a 1581 drive, it lets you open a partition, as well. Anytime a partition is the active directory, you'll see a message to that effect. You can access the root directory whenever you want by pressing the stop key. Don't deselect a 1581 drive or change disks without first returning to the root directory.

ERRORS AND OTHER THINGS

As with any disk-intensive program, there's always the possibility that an important disk or file will get lost or corrupted—through either carelessness or distraction. RUN Shell is designed to avoid such disasters through extensive error checking and verification, but, since I couldn't anticipate every user error that could occur, you should know about the following RUN Shell strengths and limitations.

All normal DOS errors are displayed in the message window during disk access (press the space bar to recover from the error), except during Copy Disk routines. If you attempt to copy a disk with track errors or a nonstandard GRC format (copy-protected), RUN Shell will just skip over the bad track, or worse, freeze up.

RUN Shell won't detect Device Not Present errors and will seem to lock up while trying to access a drive that hasn't been turned on. If this occurs, press the run-stop/restore combination, turn on the offending drive, then reinitialize the program.

Be sure you set up the logical drives according to the disk drives you actually have connected to your computer. If you have a 1541 and tell RUN Shell it's a 1581, you'll be in for an unpleasant surprise as the program tries to access track 41! If this, or something similar, happens, quickly turn off your drive or reset the computer! Likewise, don't attempt to set up a logical RAM device without a RAM expansion module connected.

All files in a RAM expansion device will remain intact unless you turn off your computer or reload RUN Shell, which resets important expansion directory pointers.

Pay close attention to which are the *source* and *target* disks during disk copies. There's nothing worse than staring at a Writing to Destination Disk message while you're holding the destination disk in your hand!

If you're a fan of using "flippies" (disks that have been double-notched to allow writing to both sides), be prepared to have occa-

sional problems with the 64-mode Copy Disk routine. The routine attempts to read track 41 of the disk to determine whether the disk is single- or double-sided (35 or 70 tracks).

Occasionally, RUN Shell will interpret a "floppie" track as a 1571 disk and attempt to copy 70 tracks of data. This won't hurt your drive; all you do is press run-stop/restore if the drive seems to stop, then re-enter RUN Shell and clean up the target disk. You should find that all the copied files have been faithfully reproduced.

RUN it right: C-64

A-Z Scramble

By Brian Melcher

ALPHABETIZING A GRID of random letters in as few moves as possible is the goal in Scramble, a strategy game that's educational for the young and just plain fun for grown-ups. Each of one to five players gets one to five grids to alphabetize by making letter swaps. When there's more than one player, they take turns.

Your score for a grid is the number of swaps it takes you to complete the alphabetizing of the letters, and your final score is your lowest grid score. The grid size for a game can range from two-by-two, with only four letters, all the way to eight-by-eight, with sixty-four letters. Duplicate letters must be grouped in order.

The program starts by asking for the number of players, the number of grids (chances) per player and the grid size. Then the game screen appears, with the grid columns lettered and the rows numbered. To swap two letters, just enter their coordinates in any order. For example, A1B2, B2A1, 1AB2, B21A, A12B, 2BA1, 1A2B and 2B1A will all make the same swap.

The area within which you can make swaps varies with the size of the grid. If it has two to four squares on a side, you're limited to adjacent letters, vertically, horizontally or diagonally; with a grid that's five to seven squares on a side, you can swap adjacent letters or those separated by one other; with an eight-by-eight grid, you can exchange over two intervening letters. You can change the swap areas by modifying line 100 in the program.

Before entering the last coordinate, you can abort a swap and get a chance to reenter the coordinates by pressing the delete key. The program checks for invalid coordinates, but doesn't count them against you; it just lets you try again.

If you decide to give up on a grid, press the £ key. You'll lose that turn, and the program will continue to your next grid, the next player or the final screen, as the case may be.

Each player's lowest score for the game so far is constantly displayed on the screen. As I mentioned, those lowest scores then become the final scores at the end of play. The game concludes with the choice of playing again or quitting.

RUN it right: **C-128 (in 40-Column mode)**

Cubix 1

By Leonard Morris

BRAIN GETTING RUSTY? You can exercise it with Cubix 1, a C-128 (40-Column mode only) three-dimensional puzzle that challenges your sense for spatial relationships and your powers of deduction. There's no need for a joystick; Cubix 1 is solved from the keyboard.

Run Cubix 1 using Menu 128 in 40-Column mode. The goal of the puzzle is to build on the screen a large cube ($3 \times 3 \times 3$ units) out of six odd-shaped sections composed of unit cubes, three sections having four each and the other three having five. These, numbered 1-6, are displayed along the top of the screen.

Each of the sections must be oriented properly, then moved into the cube construction area and positioned. To work with a section, press its number (on the main keyboard) and a larger version of it will appear on the left side of the screen. There you can rotate it, by pressing the corresponding letter keys, around the X, Y and Z axes until you have it oriented to your satisfaction so it will slide into place in the growing cube.

You fill in the big cube on the right side of the screen with the six sections. When you have a section oriented as you wish, press M to move it into the cube area, then use the numeric keypad to

position it. Numbers 1 and 9 will move it in and out along the Z axis, 2 and 8 up and down along the Y axis and 4 and 6 left and right along the X axis. Finally, you place the section to stay by pressing the P key, which redraws the cube showing the new section in place. It also causes that section's number at the top of the screen to change color, enabling you to keep track of which sections you have used.

To cancel a move into the cube—should you decide to rotate the section further, for instance—press the C key. To retrieve a section that you've already placed, press its number and then the R key. The number at the top of the screen will revert to the original color and the cube will be redrawn with the section deleted.

In addition to rotating the six sections, you can rotate the entire contents of the cube area at any time by pressing the zero key and the X, Y and Z keys.

If you attempt to do something the program doesn't allow, such as place a section in an occupied area of the cube or beyond the cube's boundaries, a tone will sound to remind you. There is no penalty. Good luck. It's not easy.

RUN it right: C-64 or C-128; joystick and 1350 or 1351 mouse

A Better Mousetrap

By Tom Brown

MANY COMMODORE USERS KNOW how to program a joystick; it's easy enough to do, even from Basic. The original 1350 mouse is equally easy, since it works exactly like a joystick. When the 1351 proportional mouse came along, however, it posed a new programming challenge: namely, to make any 1351-compatible program detect, without prompting the user for input, whether a joystick or the 1351 mouse is plugged in, and to respond accordingly. This the program can do by checking the input/output area of memory, but it's difficult to distinguish devices this way. It makes more sense for a program to be able to handle both devices simultaneously.

Interestingly enough, if you plug a joystick into the front joystick port (port 1), it will interfere with the keyboard. Therefore, most

programmers prefer to have the joystick work with the rear port (port 2). With the 1351 mouse, the opposite happens: Because of the keyboard operation, it's easier to have the mouse in the front port than in the back. Mousetrap-128 and Mousetrap-64 follow this pattern, using the front and rear ports to give the programmer access, respectively, to the 1351 mouse or a joystick—or both at the same time!

INSPECTING THE TRAP

Both Mousetrap programs are interrupt-driven machine language routines that are based in part on the original 1351 mouse reader. They use the built-in IRQ routine, which suspends what the computer is doing 60 times a second to perform housekeeping chores, such as reading the keyboard.

Mousetrap scans each port for movement and updates accordingly a pair of screen-pointer registers (one for the X position, the other for the Y). The X and Y registers contain 16 bits each, allowing for values ranging from 0 to 65535 (64K) along each axis. These values can be "scaled" to make the pointer move slower for the same amount of mouse or joystick movement.

Note that, while I refer to "pointer movement," Mousetrap simply prints the X and Y screen coordinates. To produce a visible screen pointer, C-64 users must poke those displayed values into the sprite registers, while C-128 users can employ the values directly in Basic 7.0 sprite commands.

Mousetrap lets you set upper limits for both X and Y to prevent the pointer from leaving the visible screen. The program also reads the firebutton on the joystick or the left button on the 1351 mouse with a single command.

SETTING THE TRAP

Following are descriptions of the commands for using Mousetrap. These are also documented in REM statements in the program listings.

SYS (52500)—Initializes the Mousetrap C-64.

SYS (6400)—Initializes the Mousetrap C-128.

The next four commands return information about the status of the screen pointer or input device.

A =USR(0)—The value of A is 0 if the mouse or joystick hasn't been moved since the last time this command was executed, and it's 128 (high bit set) if it has been moved. This command works a lot

faster than comparing old and new values for X and Y using Basic variables.

A=USR(1)—Returns the current X, or horizontal, screen-pointer position in variable A.

A=USR(2)—Returns the current Y, or vertical, screen-pointer position in variable A.

Note that USR(1) and USR(2) both can be scaled if necessary.

A=USR(3)—A is 128 (high bit set) while a joystick or mouse button is being pressed; otherwise it's 0. Since Mousetrap checks the button status 60 times per second, be sure to give the user time to release the button, or multiple presses will register. (Of course, these frequent checks can be handy for certain drop-down menus, which appear only when the button is being pressed and vanish when it's released.)

The following commands give instructions to Mousetrap. Since they must include a dummy variable, A (or an equivalent) must be present, but it will not return a meaningful value.

A=USR(4),X,Y—Lets you set the screen-pointer coordinates, as in **A=USR(4),50,100**. Note the peculiar syntax of this command.

A=USR(5),X,Y—As above, but allows you to set the highest possible X and Y values for the screen pointer. For example, **A=USR(5),319,199** sets the limit of pointer movement to the visible 40-column screen area.

A=USR(6),B—The value of B scales the number of pointer counts per unit of distance the joystick or mouse moves. B should fall in the range 0–7. If it's greater, it's "ANDed" with 7 to mask the higher bits and produce a value in the correct range.

The actual pointer count is divided by 2 to the power of the value in B. For example, if **B=3**, X and Y are divided by 2^3 , or 8. Be careful here, as only the values returned in USR(1) and USR(2), *not* the X and Y pointer registers, are divided in this way.

If you turn off scaling (with a value of 0), the pointer value will immediately appear in full when you next read it. Remember to reset the upper limits of X and Y appropriately if you plan to use scaling in your program.

A=USR(7)—Turns off the Mousetrap interrupt by restoring the original address in the computer's IRQ vector.

NOTES ON THE USER COMMAND

Mousetrap provides some good examples of employing the User command to interface with a machine language program. It changes

the User-command vector in the computer to point to the appropriate Mousetrap machine language routines. The command to turn off the interrupt routines does not restore the original value of the User-command vector, as that value is normally just a dummy value used by the computer on power-up.

Remember that the User command returns a value, just like any other Basic function, and that value needs to go somewhere. Either assign it to a throwaway variable, as in `A=USR(6),B`, or include the User command in an If-Then statement. If you don't, your computer may lock up, and you'll have to reset it or turn it off and on again.

NOTES ON INTERRUPTS

When you initialize Mousetrap by changing the computer's IRQ vector, the original vector is saved. Then, after each Mousetrap interrupt, execution jumps to this old vector and the built-in interrupt routine carries on. Because of this arrangement, you can "chain" Mousetrap with other interrupt routines. When Mousetrap is turned off, the original IRQ vector is restored.

PLACING THE TRAP

Lines 100-200 of Mousetrap demonstrate how to use some of the routine's commands. To use Mousetrap in your own application, just delete these lines and insert your program.

Note that, because sprites are not used, you are free to use the X and Y values any way you wish. You might use them (along with appropriate scale and Poke commands) to manipulate SID chip values or to highlight menu selections, filenames, and so forth, without sprites. You could even create point-and-click Basic programs for the C-128's 80-column screen!

Whatever your programming needs, keep in mind that, with Mousetrap, 1351 programs are always in season.

Label Base

By Bob Kodadek

LABEL BASE IS AN INTEGRATED database program for maintaining names, addresses and telephone numbers, with built-in printing routines that let you instantly print both mailing labels, in any quantity, and a personalized telephone directory. Because the program is menu-driven and its files are memory-based, it's fast and easy to move around in, and, since the record format is predefined, you can start inputting your data immediately.

What really sets Label Base apart from many other Basic programs of this type is its use of "pure" sequential files, which means that file data can be exchanged readily between Label Base and some popular programs, such as Superbase. You can even create or edit your files with a word processor that uses sequential files, such as RUN Script.

Best of all, Label Base will run on both a C-64 and a C-128 in either 40- or 80-Column mode with no modifications. In 80 columns, the program automatically takes advantage of the "wider" screen and runs in Fast mode.

CUSTOMIZING THE PROGRAM

Label Base is self-modifying, so the program can be used for both the C-64 and C-128. By default, it operates in Uppercase mode, but you can use lowercase by merely changing PS=0 in line 40 to read PS=7. This affects both screen and printer output. I believe that labels printed in uppercase have a better appearance and are easier to read; also, it's easier and faster to enter data in this mode. However, since this is a matter of personal preference, I feel it's important to have lowercase available.

Label Base is configured to print "one-across" mailing labels of the standard size, which measure $\frac{15}{16}$ of an inch wide by $3\frac{1}{2}$ inches long. If you're using a wider label, you'll need to adjust the amount of vertical space between labels by increasing variable LL in line 40 of the program.

After you've gotten the program working, type in a few sample records, then test for the proper spacing by printing them on a plain sheet of paper and placing the printout next to a strip of your labels for comparison. If the records are too close together, increase LL in line 40 and try again until you achieve the proper positioning of the text. After making this, or any other, change to the program, be sure to save the new version to disk.

USING LABEL BASE

When you run Label Base, it first determines which machine you're using and the number of screen columns available, then appropriately introduces itself as Label Base 64 or Label Base 128 and displays the main menu. If it finds itself in the 80-Column RGBI mode of a C-128, it automatically switches the 8502 processor to 2 MHz and blanks out the composite video screen.

The main menu is shown in Figure 1. To select one of the functions, press the designated function key. At any point in the program, you may return to this main menu by pressing the return key.

The Load, Save, Directory and Alphabetize functions are self-explanatory. Of course, Load and Save actually mean read and write, since Label Base files are sequential.

Note that when it's writing files, the program adds the prefix .LB to each filename. This limits the maximum number of characters in a name to 13, but enables Label Base to distinguish its files from others on the same disk and display only those files when you ask for a directory. When you're reading or writing a file, don't enter the prefix, since the program does it for you. By entering no filename, you can abort a read or write operation without affecting the current

Figure 1. Main menu.

F1 — LOAD FILE
F3 — SAVE FILE
F5 — SELECT RECORDS
F7 — DIRECTORY
F2 — ENTER DATA
F4 — ALPHABETIZE
F6 — PRINT MENU

Figure 2. Select Records Help commands.

F — FIRST RECORD
N — NEXT RECORD
P — PREVIOUS RECORD
L — LAST RECORD
K — KEY FIELD MATCH
A — ADD RECORD
D — DELETE RECORD
R — REPLACE RECORD
E — EDIT CURRENT RECORD

file in memory. The error channel is read and reported after each disk operation. To erase the current file in memory and start a new one, press F1 and enter NEW as the filename.

The Enter Data function lets you add new records to a file directly from the main menu. The blank record, or template, is displayed, complete with field names and designated field lengths, so all you have to do is enter the required data into each field and press the return key. The custom input routine automatically limits the amount of data entered.

To correct an error in the current field, use the delete key. To leave a field blank, just press return and move on. After entering the last field, press the space bar to do another record, or press the return key for the main menu. If necessary, records may be further edited from the Select Records menu described below.

While the machine language input routine will accept most characters and punctuation except the quotation mark, it's important not to use commas or colons in your data. If you do, Basic's Input# command will ignore any data after the comma or colon when the file is later read back into memory. While the Get# command will read every byte of data, I found it far too slow for reading large sequential files.

Both the Select Records and Print Menu functions provide submenus to work from.

THE SELECT RECORDS MENU

Whenever you choose Select Records, the current record is displayed on the screen, along with a menu of "quick-key" commands. Control/P instantly prints the current record on as many labels as you indicate; it can be aborted by pressing 0 and return. To find out what the one-letter, quick-key commands do, press ? to go to the Help screen, shown in Figure 2.

The first four one-letter commands instantly locate the First, Next, Previous or Last record in the current file. Key Field Match finds a record by last name, even if you don't remember the correct spelling of the entire name. In most instances, only the first one or two characters will do.

To do a Key Field search, press the K key, and, when prompted for the key field, enter the first portion of the last name and press return. The first record beginning with those characters will appear. Then use the Next Record command to quickly locate the record you want. For this to be effective, the file must be in alphabetical

order, so, if necessary, return to the main menu and sort the file by pressing the F4 key.

The remaining commands on the Select Records menu let you add new records or delete, replace and edit the current record. When you're editing a record, each field is presented individually. Edit uses Basic's own Input# routine to take data directly from the screen, so all the cursor control keys are functional. However, the Input routine doesn't limit the length of input, so be careful not to enter strings longer than Label Base will allow.

Unlike the C-128's Input# command, the C-64's won't accept an empty string (containing only spaces) from the screen or logical device #1. As a result, when using a C-64, you must represent an empty field with an asterisk on the screen, though it'll be stored as a space character. For this reason, I recommend Edit only for making minor changes to a record. Otherwise, use the Replace command to re-enter the entire contents of the record.

THE PRINT MENU

The Print menu lets you print the current record, all the records in your file or a phone directory of all the records. You'll be prompted for the number of labels you wish to print from each record, the default being 1. If, at this point, you decide not to print a label, enter 0 for the quantity and press return.

All the printing routines check for the presence of a printer with device number 4. If this printer isn't found, the command to print will simply be ignored.

The listings in the phone directory printout are formatted like those in a real phone directory and include no city, state or Zip code. The paging feature in this print routine is based on the normal page length of 66 lines.

The number of files you can input using Label Base is unlimited. However, the maximum number of records per file is 250 for the C-64 and 350 for the C-128. This should be more than enough for the average home computer user.

Even if you're used to another program for printing your labels, try Label Base. It can print a small quantity of labels so fast that you can be finished before a larger database, such as Superbase 128, is done loading.

RUN it right: C-64

Memory in Motion

By Ken Huebner

MEMORY TILES IS A COLORFUL Concentration-type game for players of ages eight to 80. It shuffles 42 rectangular tiles and displays them on the screen with their backs up. On the "other side" of each tile is a miniature computer picture.

To play, you and a friend take turns moving the cursor around the screen with the cursor keys, searching for matched pairs of pictures. Once you have the cursor over a desired tile, press the space bar, and instantly an animated dog, rocket, Christmas tree or other picture will appear. When the show is over, seek out the matching picture among the other tiles, again by moving the cursor and pressing the space bar. In time, you'll remember the positions of previously shown pictures, making it easier to find matches.

If you make a match, a melodic whistle will tell you to try for another; if you miss, a low beep will signal your opponent to take a turn. The scorekeeper at the bottom of the screen always indicates whose turn it is.

The game is over when all 42 tiles have been matched. If you want to play again—or restart a game—just press the ← key.

Now, where *did* I see that other dog?

RUN it right: C-128 (in 40-Column mode)

Thoroughbird Racing

By Hugh McMenamin

THOROUGH BIRD RACING DEPICTS A RACETRACK such as you've probably never seen before, since the entrants are two-legged and feathered. That's right; in each race, you bet on your choice

among five fleet-footed ostriches, each sporting a jaunty jockey. Wagers of \$2-\$200 on win, place, show or parlay (win and place) pay 4 to 1, 2 to 1, even and 15 to 1, respectively. Up to eight fans can wager on a race, each starting with a kitty of \$200. If Lady Luck has come with someone else, you can always borrow more from the track.

As the afternoon begins, you find yourself at the pari-mutuel window, placing your bet on the first race. The race itself is enlivened by the sound of avian feet pounding the dirt, and it concludes with an official finish photo. Your winnings and the status of your kitty are then shown, along with each bird's record for the afternoon so far.

You and your fellow fans can watch up to ten races in an afternoon, and, when you leave the track, your winnings or losses are displayed. You may want to stick your head in the sand!

RUN it right: C-64; joystick

Fast-Food Frenzy

By Colin Adams

IN KETCHUP ATTACK, a shoot-'em-up game that you'll play with relish, a barrage of hamburgers and french fries descends toward the squeezable ketchup bottle poised at the bottom of the screen.

To dispatch the incoming edibles, move the bottle left and right with a joystick in port 2 and squirt ketchup at them by pressing the firebutton. Hitting a bag of fries raises your score by 20 points, a burger by 30. Of course, while you're squirting the fast foods, you must avoid colliding with them; each collision costs a "life."

You start play with four lives, lose one to each collision, but then gain one for every level completed. To complete a level, you must splat a number of burgers equal to your level number multiplied by ten. In other words, at level two you must splat 20 burgers.

To play the game, just load and run Ketchup Attack from Menu 64.

Storybook 64

By Barry Robertson

STORYTELLING IS DIFFICULT on the computer if you want to display graphic screens along with text, but Storybook 64 makes easy work of creating a computerized storybook. You only need to know the fundamentals of editing Basic programs in order to add the words and graphic screens needed to create a story. The program supports graphic screens created by Art Studio, Advanced Art Studio, Blazing Paddles and Flexidraw.

To begin, load and run CHAPTER 1 LOADER, and it will automatically load and run CHAPTER 1. To create a story, just fill the screen that appears with your description of the first picture or an introduction into your story. To display a graphic picture you have saved on disk, press the F7 key. Press any key again and the picture will fade out to the text screen where more story text is displayed.

In the program, there are lines of GOSUBs listing the names of the graphics drawing programs in REM statements. You can place a REM in front of the GOSUBs in the lines that do not apply to your drawing programs. To access two, three or more pictures, plus text, just list lines 10-48 to the screen and change those line numbers to 50-88 for the second picture, and to 100-148 for the third. You can use line 10000 to check the program's length. If you approach 8192, just activate line 1003 (see below) to load CHAPTER 2. CHAPTER 2 can be made by saving a copy of CHAPTER 1, using the filename CHAPTER 2. Then edit CHAPTER 2 to continue the story.

CHAPTER 1 LOADER is a basic loader that pokes the machine language routines into memory, and then uses the Dynamic Keyboard method to load and run CHAPTER 1. If you don't want to show the dynamic loading, just change the text color to the screen color by adding a POKE646,X in line 36, where X is the color of the screen. (Note: The programs are not compatible with the DOS Wedge.)

After you have put some picture files on disk, you can modify lines 10-14 to load the picture file needed. Then remove the REM

at the start of line 20 to load the picture file. For Art Studio, activate lines 10, 20, 34, 40 and 46. For Advanced Art Studio, activate lines 11, 20 and 34. For Blazing Paddles, activate lines 12, 20, 22 and 36. For Flexidraw, activate lines 14, 20, 24, 38, 40 and 46. To change the text in the program, just modify lines 1074 to 1086 and add whatever you need.

The filenames now in the program can be used to load demo pictures that come with the drawing programs. The color pictures in Blazing Paddles, Art Studio and Advanced Art Studio can be transferred to the story disk as is, but the pictures in Flexidraw 5.5 must be saved as "stand alone" files that can then be transferred to this program. When you're ready to go to CHAPTER 2, just change the print statement in line 1002 to `LOADING CHAPTER 2` and remove the `END` and the `REM` from the beginning of line 1003 to automatically load CHAPTER 2. Note: At this point save the new version of CHAPTER 1 before running it.

It takes about half a minute to load a graphics file from a Commodore 1541 disk drive without using a fast loader. Therefore, enough text should be displayed to keep the reader busy during the graphics loading process. The memory locations used by this program are as follows:

2048-8191—Basic

8192-18207—Advanced Art Studio

23552-32767—Blazing Paddles

24576-32576—fade in/out routines

40960-51199—Flexidraw

53000-53300—memory movers

There you have it. Put your best collection of graphics screens on a work disk and let Storybook 64 put on a show!

RUN it right: C-128

Budget Plus

By Leonard Morris

BUDGET PLUS IS A PROGRAM designed for people who like to keep track of their finances and have records to show where all the

money went. Written for the Commodore 128, it displays data on two 80-column screens. These are switched back and forth by a technique known as page-flipping, which is accomplished by manipulating the register in the Video Display Controller (VDC).

One screen displays the monthly and yearly totals of six broad budget categories, plus a "savings" column; the other shows your budget for a given month. The latter is the screen where you do all your work, which you can then transfer to the yearly screen as monthly category totals.

The work screen also displays a menu and a calendar that shows the month and year you are currently working on. When you want to work with data pertaining to a different month, you should save the current month and then move on to the other. When the next calendar month appears, you just zero all the entries and enter the new amounts.

MENU OPERATIONS

The menu appears at the bottom of the work screen, giving you the following options to summon with one keypress:

Load: This will load from disk previously saved data for the month and year displayed on the calendar.

Save: Saves data to disk for the currently displayed month. If you've already saved a file for that particular month, it will be erased and replaced by the new data.

Zero All Entries: Zeroes all entries on the work screen and all entries for that month on the yearly screen.

X's Entry: Use this option if you want to multiply an entry by a given factor.

Add: Adds a given amount to an entry.

Yearly Report: This flips you to the yearly budget screen, from which you return to the work screen by hitting any key.

Total Year Delete: This option scratches all files for the current year beyond recovery. Make absolutely sure this is what you want before selecting this option.

Month Delete: This scratches the file only for the month currently displayed on the calendar.

Hardcopy: Gives you a printout of your budget.

Rename Item: Use this to change the current label of any sub-category on the work screen to something more appropriate. If you want the name change to be permanent, you must then use the Save option.

F1, F3, F5, F7: Used to move forward or backward through the calendar years and months.

ENTERING DATA

To make an entry, you first move the rotating cursor to the desired line; then enter your numbers or name change. You enter amounts with decimal points, in the same way that you'd write them by hand, and you can change any amount simply by typing over it. The total for each category changes automatically after you hit the return key following a new entry.

The down and right cursor keys are the only ones that move the cursor. Pressing the cursor-right key moves you to the next category of items. The cursor-down key moves you down one item line, and then, when the last item is reached, wraps around to the top of the same category.

Since I didn't design this program for someone who makes a million dollars, the highest permissible entry amount is 999,999.99; but I expect that won't hamper too many users. ■

RE₌₌RUN

EDITOR IN CHIEF

DENNIS BRISSON

TECHNICAL MANAGER

LOU WALLACE

TECHNICAL EDITOR

TIM WALSH

MANAGING EDITOR

SWAIN PRATT

SENIOR EDITOR

BETH S. JALA

ASSOCIATE EDITOR

HAROLD R. BJORNSEN

COPY EDITOR

PEG LePAGE

ART DIRECTOR

HOWARD G. HAPP

DESIGN AND LAYOUT

ANN DILLON

TYPESETTING

DEBRA DAVIES

KEN SUTCLIFFE

FULFILLMENT CONSULTANT

DEBBIE BOURGAULT

10 Programs Included on this Disk

From the July RUN:

- ▶ RUN Shell
- ▶ A-Z Scramble
- ▶ Cubix 1
- ▶ A Better Mousetrap

From the August RUN:

- ▶ Label Base
- ▶ Memory in Motion
- ▶ Thoroughbird Racing
- ▶ Fast-Food Frenzy

Plus: Extra Bonus Programs!

- ▶ Storybook 64
- ▶ Budget Plus 128

If any manufacturing defect becomes apparent, the defective disk will be replaced free of charge if returned by prepaid mail within 30 days of purchase. Send it, with a letter specifying the defect, to:

RUN Special Products • 80 Elm Street • Peterborough, NH 03458

Replacements will not be made if the disk has been altered, repaired or misused through negligence, or if it shows signs of excessive wear or is damaged by equipment.

The programs in ReRUN are taken directly from listings prepared to accompany articles in *RUN* magazine. They will not run under all system configurations. Use the RUN It Right information included with each article as your guide.

The entire contents are copyrighted 1989 by IDG Communications/Peterborough. Unauthorized duplication is a violation of applicable laws.

© Copyright 1989 IDG Communications/Peterborough

